Package 'plot.matrix'

December 9, 2019

Type Package

Title Visualizes a Matrix as Heatmap

Version 1.4

Date 2019-12-05

Description Visualizes a matrix object plainly as heatmap. It provides S3 functions to plot simple matrices and loading matrices.

License GPL-3

RoxygenNote 6.1.1

LazyData true

Suggests knitr, rmarkdown, psych

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Sigbert Klinke [aut, cre],
Frédéric Chevalier [ctb]

Maintainer Sigbert Klinke <sigbert@hu-berlin.de>

Repository CRAN

Date/Publication 2019-12-09 08:20:02 UTC

R topics documented:

`air.pvalue` *New York Air Quality Measurements*

Description

p-values of pairwise correlation test of the complete-cases of daily air quality measurements in New York, May to September 1973.

Usage

```
data(air.pvalue)
```

Format

A 4x4 matrix with p values of pairwise correlation tests (`cor.test`).

`Ozone` Ozone (ppb)

`Solar.R` Solar R (lang)

`Wind` Wind (mph)

`Temp` Temperature (degrees F)

Source

The data are derived from the New York Air Quality Measurements data set.

References

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) Graphical Methods for Data Analysis. Belmont, CA: Wadsworth.

Examples

```
data(air.pvalue)
plot(as.pvalue(air.pvalue))
```

as.cor *as.XXX conversion functions*

Description

as.XXX conversion functions

Usage

```
as.cor(x)

as.assoc(x)

as.pvalue(x)
```

Arguments

x	numeric matrix: matrix to convert

Value

a matrix with an appropriate class

Examples

```
# as.cor
c <- cor(airquality, use="complete.obs")
# as.assoc
# as.pvalue
data(air.pvalue)
plot(as.pvalue(air.pvalue))
```

assignColors *assignColors*

Description

Assign to each value in x a color according to the choices of breaks and col.

Usage

```
assignColors(x, breaks = NULL, col = heat.colors, na.col = "white")
```

Arguments

x	numeric or non-numeric vector
breaks	vector with breaks
col	vector with colors or color function
na.col	color for NA or out-of-range values

Details

Depending if x is a numeric or non-numeric vector colors are assigned to each value.

In case of a numeric vector breaks can be

- a number, giving the number of intervals covering the range of x,
- a vector of two numbers, given the range to cover with 10 intervals, or
- a vector with more than two numbers, specify the interval borders

In case of a non-numeric vector breaks must contain all values which are will get a color. If breaks is not given then a sensible default is choosen: in case of a numeric vector derived from pretty and otherwise all unique values/levels are used.

col can be either be a vector of colors or a function which generates via col(n) a set of n colors. The default is to use heat.colors.

Possible color functions in R packages can be found by vignette('plot.matrix').

Value

vector of color with the same length as x with the attributes breaks the breaks used, col the color coding and na.col the color for NA and out-of-range entries

Examples

```
## numeric vector
x <- runif(10)
assignColors(x)
# set breaks
assignColors(x, breaks=15)
assignColors(x, breaks=c(0,1))
# set colors
assignColors(x, col=c("red", "green", "blue"))
assignColors(x, col=topo.colors)
# NA and out-of-range
x[5] <- NA
assignColors(x, breaks=seq(0.5, 1, by=0.1), na.col="red")
## logical vector
l <- sample(c(NA, TRUE, FALSE), size=10, replace=TRUE)
assignColors(l)
assignColors(l, breaks=c("FALSE", "TRUE"), col=c("red", "blue"))
## character vector
t <- sample(letters, size=10, replace=TRUE)
assignColors(t)
assignColors(t, col=rainbow(5))
```

bfi.2 *25 Personality items representing 5 factors*

Description

25 personality self report items taken from the International Personality Item Pool (ipip.ori.org) were included as part of the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project. In contrast to the original data `bfi` from the library `psych` (version 1.8.12) it contains only the 25 personality self report items and the 2436 complete observations.

Usage

```
data(bfi.2)
```

Format

A data frame with 2436 observations on the following 25 variables (the q numbers are the SAPA item numbers).

`A1` Am indifferent to the feelings of others. (q_146)

`A2` Inquire about others' well-being. (q_1162)

`A3` Know how to comfort others. (q_1206)

`A4` Love children. (q_1364)

`A5` Make people feel at ease. (q_1419)

`C1` Am exacting in my work. (q_124)

`C2` Continue until everything is perfect. (q_530)

`C3` Do things according to a plan. (q_619)

`C4` Do things in a half-way manner. (q_626)

`C5` Waste my time. (q_1949)

`E1` Don't talk a lot. (q_712)

`E2` Find it difficult to approach others. (q_901)

`E3` Know how to captivate people. (q_1205)

`E4` Make friends easily. (q_1410)

`E5` Take charge. (q_1768)

`N1` Get angry easily. (q_952)

`N2` Get irritated easily. (q_974)

`N3` Have frequent mood swings. (q_1099

`N4` Often feel blue. (q_1479)

`N5` Panic easily. (q_1505)

`O1` Am full of ideas. (q_128)

`O2` Avoid difficult reading material.(q_316)

03 Carry the conversation to a higher level. (q_492)

04 Spend time reflecting on things. (q_1738)

05 Will not probe deeply into a subject. (q_1964)

The 25 items are organized by five putative factors: Agreeableness, Conscientiousness, Extraversion, Neuroticism, and Opennness.

The item data were collected using a 6 point response scale:

1 Very Inaccurate

2 Moderately Inaccurate

3 Slightly Inaccurate

4 Slightly Accurate

5 Moderately Accurate

6 Very Accurate

as part of the Synthetic Apeture Personality Assessment (SAPA https://sapa-project.org) project. To see an example of the data collection technique, visit https://SAPA-project.org or the International Cognitive Ability Resource at https://icar-project.com. The items given were sampled from the International Personality Item Pool of Lewis Goldberg using the sampling technique of SAPA. This is a sample data set taken from the much larger SAPA data bank.

Note

The bfi.2 data set and items should not be confused with the BFI (Big Five Inventory) of Oliver John and colleagues (John, O. P., Donahue, E. M., & Kentle, R. L. (1991). The Big Five Inventory–Versions 4a and 54. Berkeley, CA: University of California,Berkeley, Institute of Personality and Social Research.)

Source

The items are from the ipip (Goldberg, 1999). The data are from the SAPA project (Revelle, Wilt and Rosenthal, 2010), collected Spring, 2010 (https://sapa-project.org).

References

Goldberg, L.R. (1999) A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In Mervielde, I. and Deary, I. and De Fruyt, F. and Ostendorf, F. (eds) Personality psychology in Europe. 7. Tilburg University Press. Tilburg, The Netherlands.

Revelle, W., Wilt, J., and Rosenthal, A. (2010) Individual Differences in Cognition: New Methods for examining the Personality-Cognition Link In Gruszka, A. and Matthews, G. and Szymura, B. (Eds.) Handbook of Individual Differences in Cognition: Attention, Memory and Executive Control, Springer.

Revelle, W, Condon, D.M., Wilt, J., French, J.A., Brown, A., and Elleman, L.G. (2016) Web and phone based data collection using planned missing designs. In Fielding, N.G., Lee, R.M. and Blank, G. (Eds). SAGE Handbook of Online Research Methods (2nd Ed), Sage Publcations.

Examples

```
data(bfi.2)
library("psych")
fa <- fa(bfi.2, 5, rotate="varimax")
par(mar=c(5.1, 4.1, 4.1, 4.1)) # adapt margins
plot(loadings(fa), cex=0.5)
```

---

ind *New York Air Quality Measurements*

Description

p-values of pairwise correlation test of complete observations of the complete-cases of daily air quality measurements in New York, May to September 1973.

Usage

```
data(air.pvalue)
```

Format

A 4x4 matrix with p values of pairwise correlation tests (`cor.test`).

`Ozone` Ozone (ppb)

`Solar.R` Solar R (lang)

`Wind` Wind (mph)

`Temp` Temperature (degrees F)

Source

The data are derived from the New York Air Quality Measurements data set.

References

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) Graphical Methods for Data Analysis. Belmont, CA: Wadsworth.

Examples

```
data(air.pvalue)
plot(as.pvalue(air.pvalue))
```

plot.assoc *plot.assoc*

Description

Visualizes a association matrix with a colored or gray heatmap. As a rule of thumb the breaks are determined by the effect sizes given by Cohen (`c(-1,-0.4,-0.2,-0.05,0,+0.05,+0.2,+0.4,+1)`. You may need to modify `mar` with the `par` command from its default `c(5.1,4.1,4.1,2.1)`. See

- `vignette('plot.matrix')` for detailed examples, and
- `plot.matrix` for further parameters.

Usage

```
## S3 method for class 'assoc'
plot(x, reorder = TRUE, gray = FALSE, grey = FALSE,
  ...)
```

Arguments

`x`	matrix: association within [0,+1]
`reorder`	logical: if the rows (variables) of the loading matrix should be reordered (default: `TRUE`)
`gray`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`grey`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`...`	further parameter given to the `plot.matrix` command

Details

If either the parameter `grey` or `gray` is `TRUE` then a gray color palette is used.

Value

a plot

Examples

```
par(mar=c(5.1, 4.1, 4.1, 4.1))
# association matrix
data(Titanic.cramer)
plot(as.assoc(Titanic.cramer))
plot(as.assoc(Titanic.cramer), gray=TRUE)
plot(as.assoc(Titanic.cramer[,1:3]), reorder=FALSE)
```

plot.cor *plot.cor*

Description

Visualizes a correlation matrix with a colored or gray heatmap. As a rule of thumb the breaks are determined by the effect sizes given by Cohen (`c(-1,-0.4,-0.2,-0.05,0,+0.05,+0.2,+0.4,+1)`. You may need to modify `mar` with the `par` command from its default `c(5.1,4.1,4.1,2.1)`. See

- `vignette('plot.matrix')` for detailed examples, and
- `plot.matrix` for further parameters.

Usage

```
## S3 method for class 'cor'
plot(x, reorder = TRUE, gray = FALSE, grey = FALSE,
  ...)
```

Arguments

`x`	matrix: correlation within [-1,+1]
`reorder`	logical: if the rows (variables) of the loading matrix should be reordered (default: `TRUE`)
`gray`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`grey`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`...`	further parameter given to the `plot.matrix` command

Details

If either the parameter `grey` or `gray` is `TRUE` then a gray color palette is used.

Value

a plot

Examples

```
par(mar=c(5.1, 4.1, 4.1, 4.1))
# correlation matrix
c <- cor(airquality[,1:4], use="pairwise")
plot(as.cor(c))
plot(as.cor(c), gray=TRUE)
plot(as.cor(c[,1:3]), reorder=FALSE)
```

plot.loadings *plot.loadings*

Description

Visualizes the loadings matrix from a Factor Analysis or a Principal Component Analysis matrix with a gray or colored heatmap. As a rule of thumb the breaks are determined by `c(-1,-0.866,-0.707,-0.5,-0.4,0,+0.4,` is used. You may need to modify `mar` with the `par` command from its default `c(5.1,4.1,4.1,2.1)`. See

- `vignette('plot.matrix')` for detailed examples, and
- `plot.matrix` for further parameters.

Usage

```
## S3 method for class 'loadings'
plot(x, reorder = TRUE, gray = FALSE,
  grey = FALSE, ...)
```

Arguments

`x`	matrix: loadings
`reorder`	logical: if the rows (variables) of the loading matrix should be reordered (default: `TRUE`)
`gray`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`grey`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`...`	further parameter given to the `plot.matrix` command

Details

If either the parameter `grey` or `gray` is `TRUE` then a gray color palette is used.

Value

a plot

Examples

```
data(bfi.2)
library("psych")
par(mar=c(5.1, 4.1, 4.1, 4.1))
# Factor analysis
fa <- factanal(bfi.2, 5)
plot(loadings(fa))
plot(loadings(fa), grey=TRUE)
# Principal Component Analysis I
pa <- princomp(bfi.2)
```

```
plot(loadings(pa), digits=NA)
# Principal Component Analysis II
pa <- prcomp(bfi.2)
ld <- structure(pa$rotation, class="loadings")
plot(ld, digits=NA)
```

---

plot.matrix *plot.matrix*

Description

Visualizes a matrix with a colored heatmap and optionally a color key. It distinguishes between numeric and non-numeric matrices. You may need to modify `mar` with the `par` command from its default `c(5.1,4.1,4.1,2.1)`. For further see the vignette `vignette('plot.matrix')`

Usage

```
## S3 method for class 'matrix'
plot(x, y = NULL, breaks = NULL, col = heat.colors,
  na.col = "white", na.cell = TRUE, na.print = TRUE, digits = NA,
  fmt.cell = NULL, fmt.key = NULL, polygon.cell = NULL,
  polygon.key = NULL, text.cell = NULL, key = list(side = 4, las =
  1), axis.col = list(side = 1), axis.row = list(side = 2),
  axis.key = NULL, max.col = 70, ...)
```

Arguments

`x`	matrix
`y`	unused
`breaks`	breaks for numeric values or values for `col`
`col`	a vector of colors or a function, e.g. `heat.colors` with one parameter `n`
`na.col`	color for missing value (default: white)
`na.cell`	to draw cells with missing values (default: `TRUE`)
`na.print`	print NA (or any given characters) when values are missing. If `FALSE`, nothing is printed. If `na.cell` is `FALSE`, this will have no effect.
`digits`	number of digits for numeric data or length of string for non-numeric data
`fmt.cell`	format string for writring matrix entries, overwrites `digits`, defaults to `NULL`
`fmt.key`	format string for writring key entries, overwrites `digits`, defaults to `fmt`
`polygon.cell`	list of parameters used for `polygon` for heatmap
`polygon.key`	list of parameters used for `polygon` for key
`text.cell`	list of parameters used for `text` for matrix entries
`key`	list of parameters used for `axis`. If set to `NULL` then no information will be plotted. Instead of `key=list(side=4)` you may use `key=4` or `key="right"`.

axis.col	list of parameters used for axis for axis of matrix columns. Instead of axis.col=list(side=1) you may use axis.col=1 or axis.col="bottom".
axis.row	list of parameters used for axis for axis of matrix rows. Instead of axis.row=list(side=2) you may use axis.row=2 or axis.col="left".
axis.key	as key
max.col	numeric: if the distance between the text color and the cell color is smaller then max.col then either white or black will be used as text color, defaults to 70
...	further parameter given to the plot command

Details

A color key is drawn if either key (defaults to list(cex=1)) or fmt.key (defaults to NULL) is not NULL.

If you want to plot the matrix entries you must set either digits or fmt. For a non-numeric matrix digits gives the length of the string printed, a negative value results in right-justified string. For a numeric matrix digits determines the number of decimal places, a negative value uses a "exponential" decimal notation. You may set format strings fmt and fmt.key directly. Settings digits leads to the following format strings (n the absolute value of digits):

x numeric and digits>0:	"%+.nf"
x numeric and digits<0:	"%+.ne"
x non-numeric and digits>0:	"%+ns"
x non-numeric and digits<0:	"%-ns"

If no colors are given then the heat.colors will be used. Alternatively you may specify your own color function that delivers a vector with n colors if called by col(n). The final colors and breaks used depend if plot.matrix gets a numeric or non-numeric matrix.

Numeric matrix: In general it must hold length(col)+1==length(breaks).

1. breaks==NULL **and** col==NULL The colors are taken from heat.colors(10) and the eleven breaks are calculated as an equidistant grid between min(x) and max(x).
2. breaks==NULL **and** col **is a color function** Ten colors are taken from the color function and eleven breaks are calculated as an equidistant grid between min(x) and max(x).
3. breaks==NULL **and** col **is a vector of colors** The length(col)+1 breaks are calculated as an equidistant grid between min(x) and max(x).
4. breaks **are given and** col==NULL The colors are taken from heat.colors(length(breaks)-1).
5. breaks **are given and** col **is a color function** The length(breaks)-1 colors are taken from the color function.
6. breaks **are given and** col **is a vector of colors** If not length(col)+1==length(breaks) holds then the length(col)+1 breaks are calculated as an equidistant grid between min(breaks) and max(breaks).

Non-numeric matrix: In general it must hold length(col)==length(breaks). At first the number of unique elements in x is determined: nu.

1. `breaks==NULL` **and** `col==NULL` The colors are taken from `heat.colors(nu)` and the breaks are set to the unique elements of `x`.
2. `breaks==NULL` **and** `col` **is a color function** The `nu` colors are taken from color function and the breaks are set to the unique elements of `x`.
3. `breaks==NULL` **and** `col` **is a vector of colors** The `length(col)` breaks are calculated as an equidistant grid between `min(x)` and `max(x)`.
4. `breaks` **are given and** `color==NULL` The colors are taken from `heat.colors(length(breaks))`.
5. `breaks` **are given and** `color` **is a color function** The `length(breaks)` colors are taken from color function.
6. `breaks` **are given and** `color` **is a vector of colors** If not `length(colors)==length(breaks)` holds then either `breaks` or `color` is shorten to the shorter of both.

If the difference between polygon color and the text color is smaller `max.col` then as text color is either `white` or `black` (depending which one is farer away from the poylgon color). The distance is computed as $\Delta C/3$ as in https://en.wikipedia.org/wiki/Color_difference#Euclidean given.

Value

a plot

Note

The use of `fmt` or `fmt.key` have the same restrictions as the use of `fmt` in `sprintf`:

The format string is passed down the OS's sprintf function, and incorrect formats can cause the latter to crash the R process. R does perform sanity checks on the format, but not all possible user errors on all platforms have been tested, and some might be terminal.

Examples

```
par(mar=c(5.1, 4.1, 4.1, 4.1))
# numeric matrix
x <- matrix(runif(50), nrow=10)
plot(x)
plot(x, key=NULL)
plot(x, key=list(cex.axis=0.5, tick=FALSE))
plot(x, digits=3)
plot(x, breaks=range(x), digits=3, cex=0.6)
# logical matrix
m <- matrix(runif(50)<0.5, nrow=10)
plot(m)
plot(m, col=c("red", "blue"))
plot(m, key=NULL, digits=1)
# character matrix
s <- matrix(sample(letters[1:10], 50, replace=TRUE), nrow=10)
plot(s)
plot(s, col=topo.colors)
plot(s, digits=10)
plot(s, digits=1, col=heat.colors(5), breaks=letters[1:5])
```

```
plot(s, digits=1, col=heat.colors(5), breaks=c('a', 'c', 'e', 'g', 'i'))
# contingency table
tab <- table(round(rnorm(100)), round(rnorm(100)))
plot(unclass(tab))
# chisquare test residuals
cst <- chisq.test(apply(HairEyeColor, 1:2, sum))
col <- colorRampPalette(c("blue", "white", "red"))
plot(cst$residuals, col=col, breaks=c(-7.5,7.5))
# triangular matrix
x[upper.tri(x)] <- NA
plot(x, digit=2)
plot(x, na.print=FALSE)
plot(x, na.cell=FALSE)
```

`plot.pvalue` *plot.pvalue*

Description

Visualizes a matrix of p-values with a colored or gray heatmap. As a rule of thumb the breaks are determined by `c(0,0.001,0.01,0.05,0.1,1)` You may need to modify `mar` with the `par` command from its default `c(5.1,4.1,4.1,2.1)`. See

- `vignette('plot.matrix')` for detailed examples, and
- `plot.matrix` for further parameters.

Usage

```
## S3 method for class 'pvalue'
plot(x, reorder = TRUE, gray = FALSE, grey = FALSE,
  ...)
```

Arguments

`x`	matrix: p-values within [0,1]
`reorder`	logical: if the rows (variables) of the loading matrix should be reordered (default: `TRUE`)
`gray`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`grey`	logical: should be a gray scale color palette used or not (default: `FALSE`)
`...`	further parameter given to the `plot.matrix` command

Details

If either the parameter `grey` or `gray` is `TRUE` then a gray color palette is used.

Value

a plot

Examples

```
par(mar=c(5.1, 4.1, 4.1, 4.1))
# correlation matrix
data(air.pvalue)
plot(as.pvalue(air.pvalue))
plot(as.pvalue(air.pvalue), gray=TRUE)
plot(as.pvalue(air.pvalue[,1:3]), reorder=FALSE)
```

Titanic.cramer *Survival of passengers on the Titanic*

Description

Matrix of Cramer's V computed on the variables economic status (class), sex, age and survival of the fate of passengers on the fatal maiden voyage of the ocean liner 'Titanic'.

Usage

```
data(Titanic.cramer)
```

Format

A 4x4 matrix with Cramer's V computed on

`Class` 1st, 2nd, 3rd, Crew

`Sex` Male, Female

`Age` Child, Adult

`Survived` No, Yes

Source

The data are derived from the Survival of passengers on the Titanic data set.

Examples

```
data(Titanic.cramer)
plot(as.assoc(Titanic.cramer))
```

Index